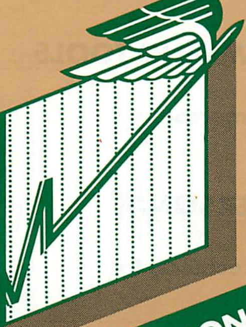
A LIBRARY OF SOFTWARE PROGRAMS FOR THE IBM PERSONAL COMPUTER

PC DISK MAGAZINE

VOL. 1 NO. 3

$29.95

USER MANUAL

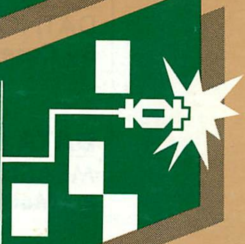


LINE PLOTTER: MAKE QUALITY LINE GRAPHS QUICKLY AND EASILY

PERSONAL CASH FLOW MANAGER—PART THREE: PREPARE YOUR RECORDS FOR SPREADSHEET ANALYSIS

SORT/MERGE I: REORDER ANY TEXT FILE IN SHORT ORDER

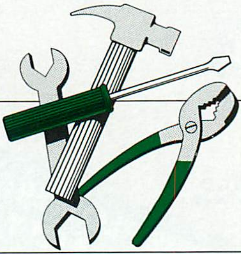
BANK IT: ENTERTAINMENT THAT DEVELOPS ESSENTIAL MATH SKILLS



DATNOIDS: SURVIVE THE ANTIMATTER MAZE WHILE BLASTING BARRIERS

CONTENTS

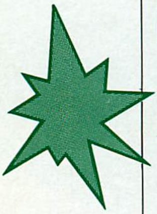
BUSINESS & FINANCE

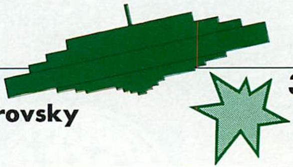


PROGRAMMING TOOLS

HOME/PERSONAL

UTILITIES & DIAGNOSTICS





GAMES

DATA FILES

DEPARTMENTS

TERMS AND CONDITIONS

LIMITED WARRANTY

In the unlikely event that the disk contained in this issue is defective in manufacture, return it within 30 days after purchase or receipt to PC Disk Magazine, 39 East Hanover Avenue, Morris Plains, NJ 07950, and it will be replaced at no charge. Merchandise for exchange must be accompanied by proof of purchase or a subscription label.

The publisher shall have no liability or responsibility to any purchaser or subscriber or to any other person or entity with respect to any liability, loss or damage directly or indirectly arising out of the use of the disk and the programs contained in this issue, including, but not limited to, any loss of business or any other incidental or consequential damages.

COPYRIGHT

The original purchaser/subscriber only is authorized to make copies solely for use in his/her own computer. Distribution, transmission, or transfer of copies to any other person is an infringement of copyright.

A LIBRARY OF SOFTWARE PROGRAMS FOR THE IBM PERSONAL COMPUTER

PC DISK MAGAZINE

PUBLISHER
André M.J. Van Hattum

EDITORIAL
Editor: Morris Effron
Associate Editor: Bohdan Shmorhay
Contributing Editors: Paul Somerson, Gary Oppenheimer, Barry Floyd, Louis Cutrona

ART & DESIGN
Cristina Botta, Linda Clenney, Robert Conrad, Don Menell Directions

CIRCULATION
Subscription Director: Chet Klimuszko
New Business Manager: Eric A. Bernhard
Renewal and Billing Manager: Shane Boel
Retail Sales Manager: Gerald Grossman

CONSUMER COMPUTER AND ELECTRONICS MAGAZINE DIVISION
President: Larry Sporn
Vice President, Marketing: J. Scott Briggs
Vice President, Circulation: Carole Mandel
Vice President, General Manager: Eileen G. Markowitz
Creative Director: Peter J. Blank
Editorial Director: Jonathan D. Lazarus
Promotion Manager: Ronni Sonnenberg

ZIFF-DAVIS PUBLISHING COMPANY
President: Richard P. Friese *President, Consumer Magazine Division:* Albert S. Traina *Executive Vice President, Marketing and Circulation:* Paul Chook *Senior Vice Presidents:* Philip T. Heffeman, Sidney Holtz, Edward D. Muhlfield, Philip Sine *Vice Presidents:* Baird Davis, George Morrissey *Treasurer:* Selwyn Taubman *Secretary:* Bertram A. Abrams

PC Disk Magazine is published 9 times per year, by Ziff-Davis Publishing Company, One Park Avenue, New York, N.Y. 10016. Subscriptions are available at $179.00 for 6 issues (residents of CA, CO, CT, DC, FL, IL, MA, MD, VT add applicable tax). POSTMASTER: send address changes to *PC Disk Magazine*, P.O. Box 2462, Boulder, CO 80322.
Editorial and Business Office: One Park Ave., New York, NY 10016. Editorial (212) 725-7770. For subscription inquiries and service write to *PC Disk Magazine*, P.O. Box 2462, Boulder, CO 80322.

ISBN #0-87165-103-3 PC DISK MAGAZINE VOLUME 1 NUMBER 3

FROM THE EDITOR

If you ask any proficient computer user how they achieved their proficiency, you may be told about the variety of excellent books they've read, or the inspirational computer guru they knew, or even the epiphany they had when it all just came together. Pursue the question a bit further however, and you'll uncover a common refrain in almost every case: they learned by doing. The fact is that where computers are concerned, there is no substitute for hands-on experience. Whether you're a beginner learning to edit a file, or a hacker working on a new operating system, the only way to confirm and expand your understanding of the computer is by trying your ideas out directly on the machine itself. And the more you use your computer, the more you will be able to use it for.

This reality of computer competence makes PC Disk Magazine a publication with unique benefits. As a regular source of ready-to-run software, it is constantly exposing you to new uses for your PC, while building up a library of software resources for you in the process. Even more significantly though, the medium of PC Disk Magazine—a diskette—requires you to use your computer in order to use the magazine. PC Disk Magazine actively encourages your use of your computer. For beginners, the action of using the magazine can be instructive in itself. For more advanced users the magazine provides tools, applications and techniques that can directly further original computer uses. In either case the very nature of PC Disk Magazine encourages learning by doing.

We intend to support PC Disk Magazine's inherent educational value by including explanatory articles on software topics in the magazine. The topics may be general in nature, such as our previous column on merging BASIC programs; or they may be related to a particular program in the issue, such as Louis Cutrona's article on recursion appearing in this issue. Some software may be even more closely tied to tutorial articles in the magazine,being present to illustrate ideas covered in the associated article.

We are also interested in providing software that is explicitly educational in nature, since the computer's capabilities as a teacher are among its most important and valuable features. This instructional software will not be confined to computer-related skills, but will range across various types of abilities, and be geared to different levels of ability as well. The program *Bank It* in this issue, which develops basic arithmetic skills, is an example of our plans in this area.

We believe that PC Disk Magazine has a unique ability to support and inspire the expanding use of computers by our readers. We intend to cultivate the opportunities of this medium in order to provide the greatest benefit to you.

Morris Effron

SUBMISSION PROCEDURE

What sets *PC Disk Magazine* apart from most other publications is that we want and need your direct involvement. That's why we developed the "Software Submission Plan." If you're interested in submitting software you've developed either on your own or with others, let us know, and we'll send you the Submission Plan booklet. We would like to give you an outline of our Submission Plan here in order to stimulate your imagination and your interest.

The Software Submission Plan provides an opportunity for software authors to profit directly from their work. Under the plan, *PC Disk Magazine* pays every published author a royalty on every issue sold which contains a copy of his or her software. Best of all, this opportunity comes without elaborate restrictions. Our desire is to license software for publication only for the disk magazine format.

To explore the considerable opportunities for publication in *PC Disk Magazine,* write to us and ask for a Software Submission Plan booklet. The address is:

PC Disk Magazine
Dept. 732
One Park Avenue
New York, NY 10016

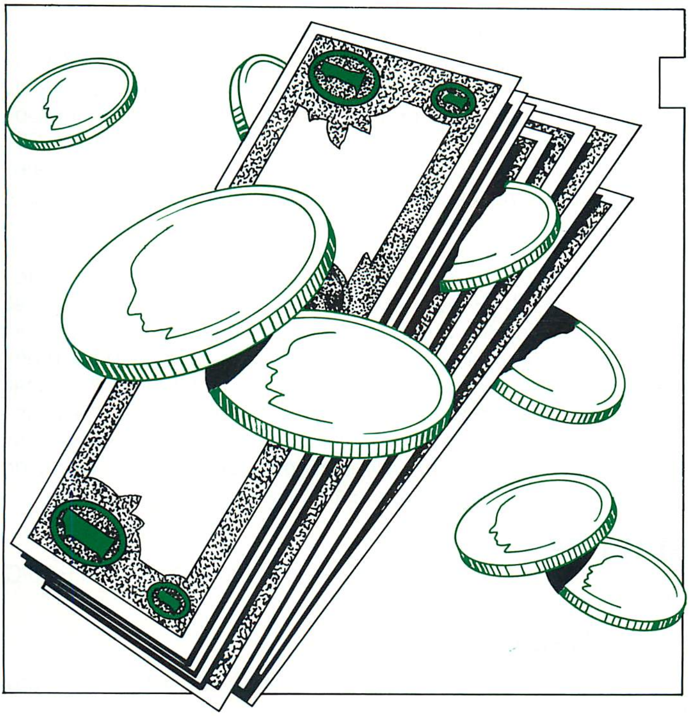
Once you've received the Submission Plan, read it over carefully. If you feel your software fits the requirements set out in the plan, complete the enclosed Software Submission Agreement and return it to us. No program code or documentation should be sent along with the Submission Agreement.

Following receipt of the Submission Agreement, the editors of *PC Disk Magazine* will evaluate the submission. At that point, we'll either indicate a lack of interest in the software or, if we are interested, we'll issue a submission authorization number and ask you to provide us with an executable copy of the program along with documentation on 5¼″ diskettes. Again, no material should be sent until you receive a submission authorization number.

If we subsequently decide that we would like to publish the software in *PC Disk Magazine,* we will offer a Software Contract, which will include such items as royalties, advances, and program and documentation changes required (if any). You will be asked to complete and test any program modifications agreed to in the Software Contract, and *PC Disk Magazine* will conduct a formal validation of the program and documentation.

We estimate that this process, from our initial evaluation to publication, takes approximately 3 to 6 months. This estimate is dependent upon a number of factors, and the process may take more or less time for your submission.

We look forward to hearing from you.



DEPRECIATION PAK

By Software Integrated Systems

Special Requirements: Printer
Files Used: DP.BAS
DEPAST.BAS

Whether you're sorting through the fiscal ramifications of alternative depreciation methods, or just keeping depreciation records for your assets, you'll appreciate the ease and efficiency with which DEPRECIATION PAK *calculates the depreciation values of the assets you own or plan to own.*

DEPRECIATION PAK *will store the characteristics of any asset and calculate its depreciated value for a given month or year, or print a detailed depreciation schedule, period by period, as calculated over the useful life of the asset.*

You'll find DEPRECIATION PAK *invaluable in helping you determine whether to employ a standard or accelerated depreciation schedule for tax purposes; whether a so-called ''shelter'' is really a benefit or a liability; or whether to sell an asset this year or next.*

Whether you're a broker, investor, or merely one who wants to make the most of his assets, you'll agree that this software is one asset that doesn't depreciate in value.

BACKGROUND

DEPRECIATION PAK can employ any of seven depreciation methods: straight line, 125% declining balance, 150% declining balance, 175% declining balance, 200% declining balance, sum of the years' digits, and Accelerated Cost Recovery System (ACRS). Within ACRS, *DEPRECIATION PAK* supports a submenu of five alternative methods: 3-year property, 5-year property, 10-year property, 15-year real property, and 15-year low income property. *DEPRECIATION PAK* works from an asset file it helps you to create. The asset file contains asset records, each of which contains descriptive information about a particular asset. You can create more than one asset file if you want to segregate your asset records.

SEVEN METHODS ARE AVAILABLE TO DEPRECIATE ASSETS

For any asset file there are two ways in which you can access depreciation data: by individual asset record or by a group of asset records. By using the group (collective) access method, you can ascertain the depreciation values of all assets that have been entered in a particular asset file. By using individual depreciation (that is, individual asset depreciation) you can not only determine an asset's depreciation value for a particular month or year, but you can also calculate a complete depreciation schedule on a monthly or yearly basis for the individual asset.

START UP

To start *DEPRECIATION PAK,* you must first load Advanced BASIC into your PC by typing:

```
BASICA ↵
```

Then place your *PC Disk Magazine* diskette in the default drive and type:

```
RUN“DP ↵
```

A title screen will appear, followed by the main menu.

ALLOCATING DATA SPACE

Your first step is to initialize an asset file. In effect, you're naming and allocating space for a data file which will hold your asset records (asset descriptions). To initialize an asset file, select option 1 from the main menu by typing:

```
1 ↵
```

The next screen will ask you to type in the asset file name. Having named your file, press the Enter key and the screen will ask how many asset records you want to reserve space for in this asset file. Once this number is set for a file it cannot be changed. However, don't be too generous in allocating space—each asset record you allocate requires 128 bytes. (Should you wish to check your remaining space, return to DOS, then run CHKDSK.) After answering the number of assets prompt you will return to the main menu.

Note that the main menu screen also lists the asset file which is currently active. Once you have created more than one asset file, use option 2 of the main menu (Select File) to switch from one asset file to another during your work session. Asset records are only accessible from one selected file at a time.

DEFINING ASSET RECORDS

To either initially load or subsequently update the records of an asset file, select option 3 of the main menu and enter the required data for each of the eight asset features in an asset record. The first, an asset ID, consists of from one to five numbers or capital letters. Next, the name of the asset may be up to 35 characters. The third line, Depreciation Method, has its own submenu of choices. Press:

↵

once you get to this line and the program will put you into the Depreciation Methods Submenu. This menu has 11 choices, the last five of which are the ACRS (Accelerated Cost Recovery System) options. Type in the number of the depreciation method desired, and follow that by pressing the Enter key. You will be returned to the asset record update screen, where your choice of depreciation method will now be displayed.

The date of acquisition must be entered in the form month/day/year formatted as MM/DD/YY. The cost of the asset must be under $10 million and greater than zero (dollar signs and commas are optional). Enter the salvage value (0 is acceptable, negative values are not), then enter the useful life in years (up to 99).

Certain depreciation methods do not allow a salvage value. In their case this item is automatically filled as N/A and skipped over (you can not subsequently edit it either). Similarly, the choice of an ACRS depreciation method fixes the useful life, so this item will be automatically set to the appropriate number of years. Lastly, choose whether or not the asset is to be included in the group depreciation activities. The advantage of marking an asset record as a group record is that it will be automatically included when you request the sum of current depreciated values for all your group assets, based on each asset's stage in its depreciation. Every asset can also be accessed individually, whether marked as group or not. With individual access you can print an entire depreciation schedule for a particular asset.

YOU SPECIFY WHETHER OR NOT TO INCLUDE EACH ASSET IN GROUP REPORTS

PRINTING THE DATA

DEPRECIATION PAK can print the contents of an asset file, asset by asset. To do this, first select the asset file containing the asset record(s) in question by using option 2 of the main menu and typing in the asset file's name. To simply print out all of the asset record data in the selected asset file, choose option 4 from the main menu and follow the directions to prepare the printer. This is useful for verifying the accuracy of the asset record data in an asset file.

But now let's have *DEPRECIATION PAK* do some real work. Options 5 and 6 from the main menu allow us to calculate and print the depreciated values of our assets either singly or in groups.

GROUP ASSET DEPRECIATION

To depreciate assets in a data file as a group, press:

5 ↵

from the main menu. Following the submenu's choices, specify whether *DEPRECIATION PAK* should calculate a depreciation value for a specific month or a specific year, and enter the date for which you want your collective asset depreciation values. Prepare your printer according to the prompts. *DEPRECIATION PAK* will then print the depreciated values of each group asset in the data file (those you marked as group assets in their asset records) and aggregate the results.

INDIVIDUAL ASSET DEPRECIATION

If you are analyzing the depreciation values for only one asset at a time you will do best to depreciate your assets individually. To do this, select the data file containing the asset in question, then press:

6 ↵

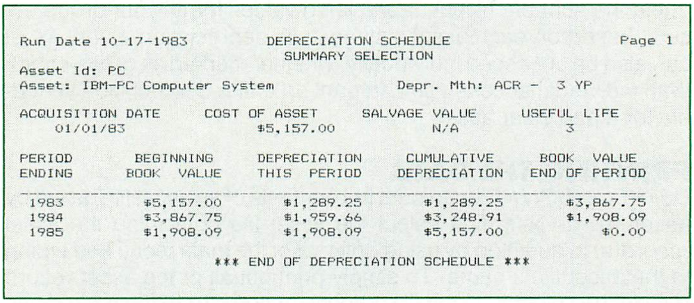
PRINT A COMPLETE MONTHLY OR YEARLY SCHEDULE FOR ANY ASSET

from the main menu. Follow the submenu's choices, choosing to depreciate for a specific month or a specific year, or choosing to print a complete monthly or yearly depreciation schedule over the useful life of the asset. After choosing the desired report, follow the screen's prompts to prepare the printer. Then type in the asset ID and press the Enter key. When the report is done you will be prompted to enter another asset ID for reporting. If you just press:

↵

the program will print the same report for the asset in the file that immediately follows the one just reported. (In this way you can print a report for all your assets by entering the first asset ID when prompted, then pressing the Enter key after each report is completed to select the next asset automatically). To exit this report mode and return to the reporting submenu, press:

Esc



```
Run Date 10-17-1983          DEPRECIATION SCHEDULE                   Page 1
                               SUMMARY SELECTION
Asset Id: PC
Asset: IBM-PC Computer System                    Depr. Mth: ACR -  3 YP

ACQUISITION DATE     COST OF ASSET     SALVAGE VALUE     USEFUL LIFE
    01/01/83              $5,157.00          N/A              3

PERIOD          BEGINNING       DEPRECIATION       CUMULATIVE       BOOK  VALUE
ENDING         BOOK  VALUE      THIS  PERIOD     DEPRECIATION     END OF PERIOD
---------------------------------------------------------------------------------
  1983          $5,157.00         $1,289.25        $1,289.25         $3,867.75
  1984          $3,867.75         $1,959.66        $3,248.91         $1,908.09
  1985          $1,908.09         $1,908.09        $5,157.00             $0.00

                     *** END OF DEPRECIATION SCHEDULE ***
```

Yearly depreciation schedule for IBM-PC hardware under ACRS 3 year method.

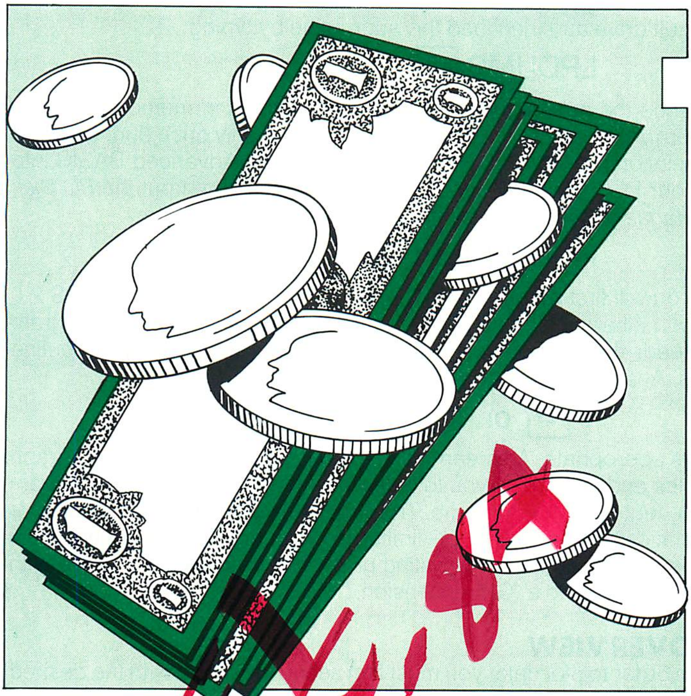
EXITING

To return to the main menu from any point in the program, press:

Esc

repeatedly to back up to the next highest menu level until you reach the main menu. Then press the Escape key again to exit to DOS. The Escape key will also interrupt the printing of any report and return you to the most recent menu.

BUSINESS & FINANCE



LINE PLOTTER

By Barry Floyd

DEL 3-15-87

Special Requirements: Color/Graphics Adapter
Graphics Printer for Hardcopy
Files Used: LINEPLOT.BAS
LPDUMP.COM

Do you have a mass of data that needs to be plotted in order to make sense? Are you searching out trends or relationships among seemingly unrelated information? Then try LINEPLOT*—a program that allows you to plot up to four separate sets of data as four separate lines, and has axes scales you can modify to expand or compress your results. You enter your data into a file using a simple full-screen procedure that allows you to label each individual X-Y data point. Each separate set of data points has a separate data screen. Once you've loaded up to four data sets, a second screen form allows you to set up your graph. Fill in the fields that define your axes labels and limits, then fill in the data set titles. Press a button and* *presto! the plotted and labeled data appear on your screen as a polished line graph.*

START-UP

Before running the plotter itself you need to load a subroutine that enables the computer to dump a copy of your line graph from the screen to your printer. Put the *PC Disk Magazine* diskette in the default drive and then load this subroutine by typing:

LPDUMP ↵

LOAD A SUBROUTINE FOR PRINTING BEFORE RUNNING THE PROGRAM

Once the subroutine is loaded, it will remain in memory until you turn your computer off, so you need load it only once during a work session. To start *LINE PLOTTER* itself, load Advanced BASIC into your PC from one of your system diskettes. Then put the *PC Disk Magazine* diskette back in your default drive and type:

RUN"LINEPLOT ↵

You will first be asked whether you are running DOS 1.1, DOS 2.0 with a hard disk, or DOS 2.0 with a floppy disk drive. The program needs this information to know where the graph printing subroutine is. Answer by typing:

1 ↵ or 2 ↵ or 3 ↵

as appropriate. A screen will then come up listing all existing graph files and prompting you to enter a filename. Note that the program accepts only the filename. The file extension (.GRF) is added automatically. Filenames are limited to 8 characters. Entering a file name not listed as an existing graph file will create a new file with that name and a .GRF extension.

OVERVIEW

In order to plot data, you must first set up the graph with the desired axes labels and limits. Once the filename has been entered, the program will prompt you to select either F5 (Set-Up Graph) or F6 (Edit Data). Selecting F5 (Set-Up Graph) will bring you to the Set-Up Graph screen. Once the parameters for setting up the graph have been entered, data points can be entered and modified using the F6 Edit Data function. Each set of X-Y data points can also be assigned a label. Once the graph has been set up using F5, and the data points have been entered using F6, the line graph itself can be generated using the function key F4 (Graph Data). The remaining three sections will describe the details of setting up the graph, entering the data, and producing the line plot.

SETTING UP THE GRAPH

The Set-Up Graph menu contains prompts for the screen label, axes labels, and axes scales. The "Top of Screen label" sets the title appearing at the top of the graph screen. The "Side of Screen label" is used to label the Y-axis, while the X-axis will have two label lines at the bottom of the graph ("Bottom of Screen label" and "Second 'Bottom' label").

Below the four screen label prompts are two yes or no questions concerning whether to display or suppress the labels and the graph grid. Your answers to these two questions respectively determine the presence or absence of all labels and the graph grid on your resulting graph screen. Try plotting with different yes and no settings to see their effect.

The next two prompts deal with the lengths of the X and Y axes. These parameters determine the size of the line plot proper. With both at 100, the line plot is at its maximum size—100% of the possible size in both the X and Y directions. Any value below 100 will reduce the length of the appropriate axis to that percentage of its maximum length, making a complete but smaller graph.

At the end of the Set-Up Graph menu are prompts for up to four data set titles. These data set titles will automatically appear at the bottom of your graph screen as a legend which associates each line of the graph with the name of the data set plotted.

The bottom two lines of the Set-Up Graph menu screen list the function keys. The F3 key is used to enter the various parameters on the screen and to alter them. The F6 key is used to enter Edit Data mode, where you create and modify the data points of your graph. F4 is used to graph the results of the F5 and F6 activities. Key F9 closes the graph you are currently working on and lets you initiate work on another. Key F10 is used to exit the program.

To fill in the fields of the Set-up Graph menu, use the up and down arrow keys (cursor control keys) to go from one line to the next. When the position indicator is located at an item you wish to enter or change, press:

F3

to open that item for editing. Single quote marks will appear as an indication that the item is open for editing. Type in the information you want, typing over the current contents as appropriate. When you have entered the value you want, press:

↵

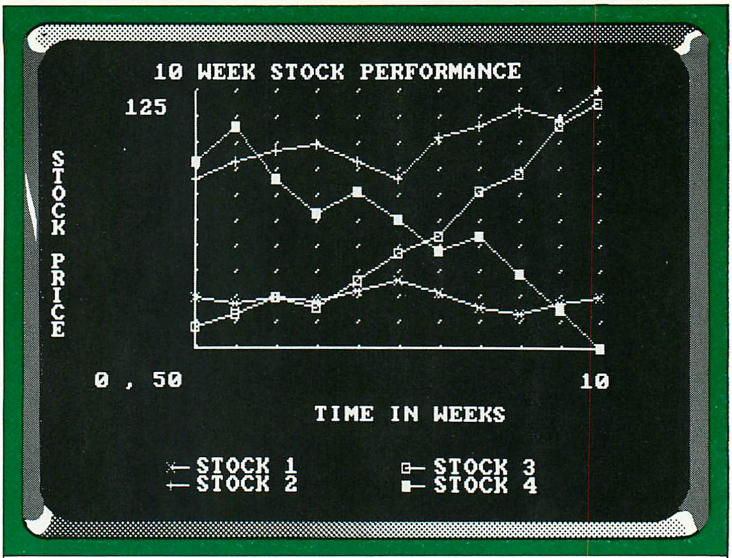
and the program will accept the new value.

If you make a mistake in entering labels or values in the Set-Up Graph menu, use the cursor control keys and key F3 (Alter Item) to go back and correct your mistake. Once you have set up all the necessary parameters for your graph, use the F6 (Edit Data) function key to load data into your file.

YOU MUST ENTER YOUR DATA AND SET YOUR GRAPH FEATURES

ENTERING AND PLOTTING THE DATA

Data points are entered and plotted in sets. One data set constitutes a single line. *LINEPLOT* allows up to four such data sets to be specified. The Edit Data screen lists the data points by label. Use the cursor control keys (here the left and right arrow keys can be used as well) and F1 (Add Item), F2 (Delete Data), and F3 (Alter Item) to input, remove, or change the data values and labels appearing on the screen. Once all of the data points in a particular set have been entered, use the function keys F7 (Prior Set) and F8 (Next Set) to move into different Edit Data screens containing other data sets. Once all data points for all data sets have been entered, the data points can be plotted by pressing:



Sample graph—should have shorted Stock 4.

When the graph itself is displayed, there is no function key legend at the bottom of the screen telling you which keys perform which functions. You must remember that the following options are available to you when the graph is displayed:

F5

returns you to the Set-Up Graph screen, as is true throughout the program.

F6

returns you to Edit Data Mode, also true throughout the program.

F10

will exit the program. Pressing this key at any time during program execution will exit the program and leave you in Advanced BASIC.

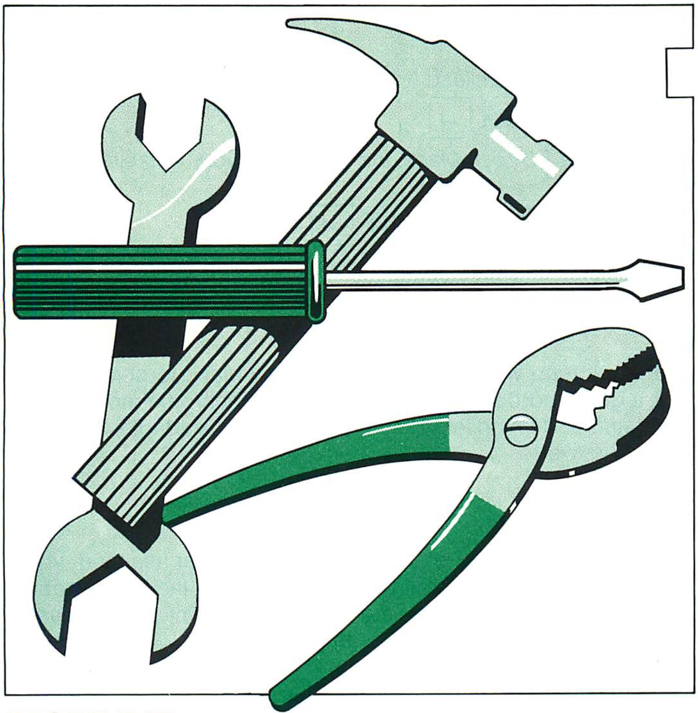
F1

will make a printed copy of your graph on any IBM-like printer with the graphics feature.

F9

will make a printed copy of your graph on any Okidata printer. Both print functions will leave you with the graph screen still displayed, so after printing press one of the other three active function keys to proceed.

YOU CAN PRINT YOUR GRAPH WHEN IT IS DISPLAYED (WITH THE RIGHT PRINTER)



BIGPRINT

By Paul Somerson

Special Requirements: None.
Files Used: BIGPRINT.BAS

Ever try to make a title screen or wall poster with your PC? Some dot matrix printers can handle type that is wider or bolder than usual, but no matter how dark or how fat the characters, you can't vary the height. BIGPRINT *will let you create G-I-A-N-T messages on your screen, and copy them to your printer. And, you can save the screens on a disk for use in your own programs.*

BACKGROUND

BIGPRINT is designed to work on either a color or monochrome monitor. (However, there are several flashy features that will work only on a color system.) It doesn't require a printer, but if you want to create large-type signs, title pages, and notices, you'll need one. The program will work with virtually any kind of printer.

BIGPRINT has a number of features that must be set at program initialization. These include the choice of display characters and colors, as well as whether screen images are to be saved to a disk file manually or automatically. All of these saved screens can be printed. Saved screen images can also be automatically chained together by a master screen-loading program that *BIGPRINT* will create at your request. These procedures will be detailed below.

START-UP

To start the program, load Advanced BASIC into your PC by typing:

BASICA ↵

Then put the *PC Disk Magazine* diskette in your default drive and type:

RUN"BIGPRINT ↵

The first screen will ask if you are using a color monitor. If you are, type:

Y

THE START-UP QUESTIONS CONTROL PROPER DISPLAY, PRINTING AND SAVING

and the screen will change from white-on-black to blue-on-white with a green border. If you are using a monochrome monitor, or a black and white monitor with the color/graphics adapter card, you should answer:

N

to this question.

You'll next be asked whether you have a printer attached, and if so, whether it can accurately reproduce all of the PC's graphic characters. If you don't have a printer that can print all 255 characters, and you try to fool the program, you'll end up with giant letters made out of things like parentheses. If your printer lacks this graphics capability, and you truthfully type in an "N", the program will create large characters out of asterisks or normal size characters, which looks far better.

Once you've answered the questions concerning your color monitor and printer capabilities, the program will ask you whether you want instructions. Answer with "Y" to this prompt to get an explanation of the seven function keys used to operate *BIGPRINT* (these keys are also described in greater detail below).

Whether you choose to read the function key instructions or not, the program next displays information about using the Escape key to restart screens on which you've made a mistake. Pressing:

Esc

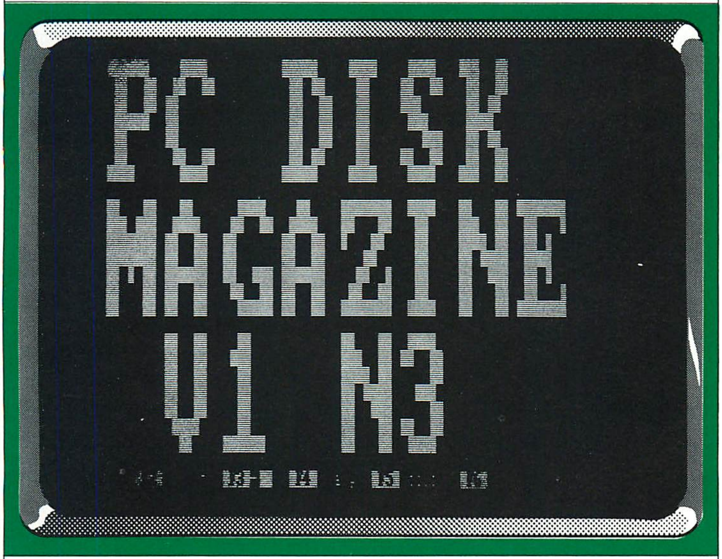
allows you to cancel your current working screen and start over.

The program also tells you at this point the number of bytes required for each screen image and queries you whether or not you want the program to create a master screen-loading program by automatically saving up to 40 screens for later viewing. If you answer "Y" to the query, the program will further prompt you concerning which of three screen change rates you prefer.

The first method of displaying screens loads all of your screens in quick succession, without any delays, and provides a sort of quasi animation. The second method loads screens one at a time, and won't load the following one until you hit any key on the keyboard. (Don't forget to hit a key; there is no on-screen prompt to remind you.) This gives you the equivalent of a slideshow at your own pace. The third mode loads the screens automatically, with a delay of about five seconds between each.

Once your screen changing procedures have been determined, the final step of the start-up sequence is to determine which of eight graphics characters you want to use to create your text. Note also that the characters can be used as building blocks to generate the larger versions of themselves (Selection 9). Choose one of these selections and begin typing your message onto the blank screen that follows.

DISPLAY CAN BE SAVED SINGLY OR AS A CHAINED GROUP



PC DISK
MAGAZINE
V1 N3

A quick and easy title screen.

OPERATION

Once you begin actually writing onto screens, everything in this program is controlled by function keys. This means that the program will work in Advanced BASIC only. The actions associated with the function keys are as follows:

F1

lets you end the program at any time. It's a good idea to end it this way rather than by hitting CTRL-BREAK, since using this key will print out a status report at the end, telling you how many files you've saved to disk, what they're called, what files you'll need to display them, and what you've named your master screen-loading program.

F2

copies the current full screen to your printer, except for the bottom line which displays the function key settings.

F3

selects the character building block that will be used to print out the giant letters and numbers on the screen and on your printer. You may choose among 9 different building blocks. Each time you hit this function key, a new building block is displayed on the bottom line of the screen. You can keep cycling through them to see which one you want to use next; when you get to the end of the list, the program will cycle around to the top of the list of choices again. Note: ?? in the F3 field of the bottom line means that the program will create a larger version using the alphanumeric character you type (equivalent to choosing Selection 9 when first entering the program).

THE FUNCTION KEYS CONTROL ALL ASPECTS OF DISPLAY CREATION

F4

(color only) toggles between single-color and multi-color letter display. If you have a color system and you hit this key, the large characters will be created out of building blocks of multiple colors. To switch back and forth between single-color characters and multicolor characters, just tap this key. As with most toggles, hitting it will switch from one state to the other.

F5

(color only) switches among different single colors. It shows you on the bottom line of the screen what color you'll be using to create the large characters. Like the building block selector F3, this cycles automatically through a fixed list.

F6

(color only) switches border colors. While the screen image you will save will not include the border color, you can nevertheless change it as you create each screen just to make the process of creating screens easier on the eyes.

F7

will save any screen to a disk file. If you so specify at the beginning of the program, *BIGPRINT* will save and name up to 40 screens on your disk *automatically* by creating a master screen-loading program on your disk (after asking you what you want to name this program). If, however, you want to create just one or two screens, and give them special names, this key will prompt you for the name and then save the screen to your disk.

If you start creating a screen and make a mistake, you can erase the previous character, but only the previous character, by hitting the backspace key. If you make several mistakes, and want to start over, you can do so by pressing:

Esc

which will clear the screen and put you back in the upper lefthand corner again.

Each screen can hold three rows of ten characters each, for a total of 30 characters per screen. If you want to center giant characters, remember to hit spaces to fill the empty onscreen positions. Type in several letters. Try your name. Then make sure your printer is turned on and press F2 to copy the screen to your printer. Then try pressing F7 to save the screen to your disk. If you had not asked earlier for automatic screen saving, at this point a prompt will ask you what you want to name the screen on your disk. After *BIGPRINT* saves the screen it will clear your monitor's screen and let you start to create another screen. (You don't have to copy the screen to the printer each time you save it to the disk.)

After you've created several screens, and saved each separately with the F7 key, press F1 to quit. If you requested a master program, you can see all the screens you've saved simply by pressing the "R" key. If you want to quit, hit any other key.

USING YOUR SCREENS

To view screens that you've saved one at a time, simply enter the following program in BASIC or Advanced BASIC:

For monochrome screens:

```
10 DEF SEG=&HB000
20 BLOAD "FILENAME.PIC",0
```

For color screens:

```
10 DEF SEG=&HB800
20 BLOAD "FILENAME.PIC", 0
```

USE BIGPRINT RESULTS IN YOUR OWN PROGRAMS

In either case, substitute the name you gave each screen where "FILENAME.PIC" is shown above. (*BIGPRINT* automatically adds the ".PIC" extension when it saves screens).

To display all the screens with the master screen-loading program, make sure the relevant .PIC files are all on disk, then run the master program. If you have named your master program SLIDE.BAS then load in Advanced BASIC and type:

```
RUN "SLIDE
```

This program can be incorporated into your programs to provide dazzling titles or messages. In order to do this, save the master program in ASCII format by typing:

```
LOAD "mastprog

SAVE "mastprog.ASC",A
```

where mastprog is the name of your master program. Then make sure this saved file is on a disk in your default drive, and load your own original program. Type in:

```
LOAD "myprog

MERGE "mastprog.ASC
```

and the master program will automatically be added to your own. Then remember to save your own program with the master program merged in. If you do want to incorporate it, make sure you renumber the master program if necessary so it doesn't replace portions of your original program. Do this before you save the master program in ASCII format by using the RENUM command.

PROBLEM RECOVERY

There really is not a great deal to say about problem recovery with *PC Disk Magazine*. If you use this software on the right equipment (IBM PC or PC-XT) running the appropriate system software, (DOS 1.1 or 2.0, Microsoft Advanced BASIC 1.1 or 2.0) you should experience no problems. Nevertheless, a few comments may resolve some more obvious difficulties.

Any BASIC program can be interrupted at any time by pressing:

[Ctrl] [Scroll Lock]

If you do not see the OK message immediately, indicating that you are back in BASIC, press these keys again. This is a rather drastic but effective way of regaining control of the computer. You won't damage any of the programs in this way, since they're still intact on the diskette. However, you will lose any data you entered while the program was running.

If you interrupt a BASIC program you may find that the function keys no longer perform as they had before starting the program. This is because many *PC Disk Magazine* programs reset the function keys during execution, then restore the original settings upon completion. An interrupt causes an abnormal termination of a program, so the function keys are not restored. To correct this situation, simply exit from BASIC and then return to BASIC.

You may find at times that the cursor control keys are not working as they should. This is because the keys are not in cursor control mode. They key that switches these keys between numeric mode and cursor control mode is the Num Lock key. So to restore the keys to cursor control mode press:

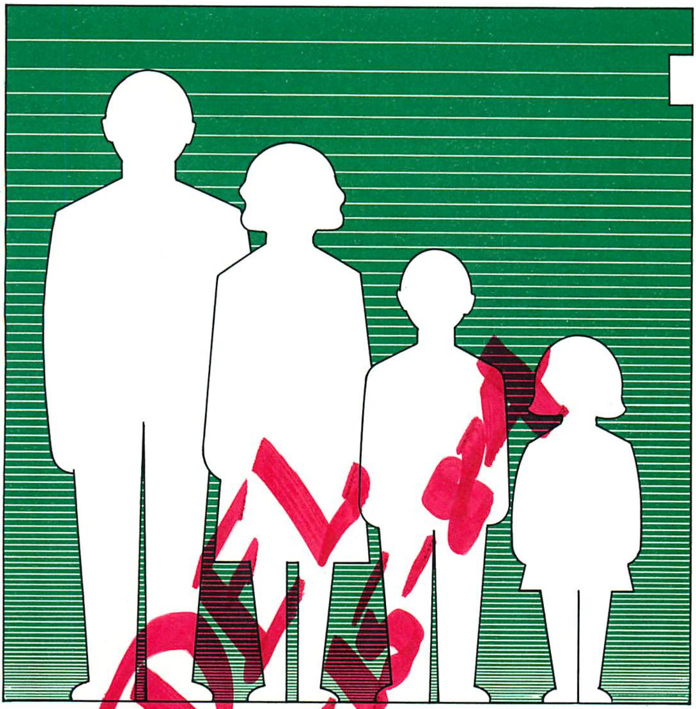
[Num Lock]

If you try to send something to the printer when there is no printer, or when the printer is off or offline, you will hang the system. The computer will just sit there and will not respond to any keys pressed. After a few seconds you may get a BASIC error message indicating that the device was unavailable. The program that was running has been aborted, and you will be left in BASIC. If the computer does not put out any message, but just remains hung, you will have to say goodbye to whatever you were doing and re-boot your system.

Though we hope you will never need it, if you should find a "bug" in a *PC Disk Magazine* program, the address to write to is:

PC Disk Magazine
Dept. 741
One Park Avenue
New York, N.Y. 10016

HOME/PERSONAL



DELETE 3-1

BANK IT

By Psychotechnics Inc.

Special Requirements: Color/Graphics Adapter
Files Used: BANKIT.COM
MONTM
BANK

Here's a fun game to help your child increase his or her basic math skills. BANK IT *by Psychotechnics Inc. combines the excitement of a "race against the computer" with amusing color cartoon graphics and a structured approach to learning. The program builds step-by-step as your child responds to each question.*

BACKGROUND

Here's how it works: when starting the program, the student is asked to choose certain color and sound options. Then the student enters a code for his or her initials. Each of these steps develops skill and ease in using the computer to find and enter information. Once the preliminary questions have been answered, the student presses the Enter key to play. Then, the race is on!

THE FIRST PLAYER TO BANK $10 IS THE WINNER

When the first problem appears on the screen, you will see two piggy banks along with a field displaying coins and bills in various combinations. The object of the game is to determine the total amount of money in the field within the time allotted. The student must use the numeric keypad to enter the answer, and remember to include the decimal point. If the student enters the correct answer before "Time's Up!" flashes on the screen, the money is deposited in his piggy bank. If he takes longer than the time limit, or if his answer is not correct, the money goes into the computer's piggy bank. The first player to bank $10 is the winner.

A unique feature of *BANK IT* is the special hidden teacher/parent option. This option allows the allotted response time to be increased or decreased. In this way, you can monitor the student's progress and increase the speed of the game along with his increasing skill. This feature also insures that the student will not become frustrated by a program which proceeds faster than he can reasonably handle.

Lively color graphics reinforce the learning as well as the fun of *BANK IT*. And, the language used is encouraging and nonjudgmental.

START-UP

To start *BANK IT,* you must be in DOS. Then put your *PC Disk Magazine* diskette into the default drive and type:

BANKIT ↵

After the Telemath title page appears, you will be asked:

Do you want color?

As the message at the top of this screen indicates, all numeric replies in *BANK IT* from these set-up questions to playing the game itself should be made using the numeric keypad on the right-hand side of your keyboard. In addition to the keys for the numbers 1 to 9, you should use:

0 Ins

for zero, and:

. Del

for the decimal point.

Returning to the question about color, if you have a color monitor you can respond by typing:

1

If you have a black and white monitor, you should type

2

to proceed.

The next question which appears is:

Do you want sound?

If you would like a little musical encouragement to begin the game, type "1" for yes or "2" for no as with the color question. After the two introductory questions have been answered, the main menu for *BANK IT* will appear. You will see two options:

F1 = Help
ENTER = Play

To learn the rules of the game, press:

[F1]

The help screen will give you a quick review. Once you have reviewed the rules, press:

[↵]

to begin to play *BANK IT*. If you do not need the help review, press:

[↵]

to proceed directly to the play of the game.

In addition to these two designated options, a special hidden teacher/parent option is available when this main menu is displayed. To use this option, type:

7714

when this main menu is displayed. No following Enter key is necessary and no numbers will appear (echo) on the screen.

YOU SET THE ALLOTTED PROBLEM RESPONSE TIME

The teacher/parent option screen that then appears will indicate the program's current student response time in seconds. The standard response time for *BANK IT* is 30 seconds. Using the numeric keypad, you can type in a new number to either increase or decrease the number of seconds allotted to solve each *BANK IT* problem. Press the Enter key to register your new time setting. Once a new response time has been chosen, it will be in effect until you end the current run of the program. If you wish to leave the current response time as is, just press:

[↵]

This feature allows you to adjust the program to the student's skill level while allowing him the independence of playing the game himself. Once a time is set, the program will automatically return to the main menu.

Before the game begins in earnest, the student must name his piggy bank. The last screen to appear before beginning play displays a chart listing the numbers 1 through 26. Each number corresponds to a letter in the alphabet. The student will be asked to pick the number beside his first initial and to enter that number, using the numeric keypad; then to repeat the process for his last initial. This exercise provides a "warm-up" to help the student recognize and enter numbers from the keypad.

PLAYING THE GAME

The following game screen displays two cartoon piggy banks—and one of them has the student's initials right under it! The other piggy bank is called "Me". This is the opponent, the computer. Above the piggy banks the student will see a field displaying coins and bills. He must quickly add up the money and enter it in his piggy bank before the computer flashes "Time's Up!"

If he beats the computer, the money trickles down into his piggy bank and the screen flashes "That's Right! Get ready for next problem".

If he gives an incorrect answer, the money goes into the computer's piggy bank and the screen flashes "You Missed! Right answer is:- - - - - - - - - - ". The correct amount is displayed, and then the program proceeds to the next problem.

The game continues until the student's piggy bank or the computer's piggy bank contains at least $10. At the end of the game the winner's piggy bank flashes on the screen announcing "(player's initials) is the Winner!" This message will keep flashing until you press:

Esc

to return to the main menu for another game.

EXITING

The Escape key can be used at any time during your use of *BANK IT* to go back to the previous level of activity. Each time you press the Escape key, you go back one level in the start-up sequence. For example, if you decided not to have sound, but now you and the student would like a little fanfare, just press:

Esc

repeatedly until the color option question appears on the screen. Similarly, you may want to change the response time once the game has started. Press the Escape key once to cancel the game in progress and return to the main menu. From there you can use the hidden time option to reset the response time, then begin play again.

THE ESCAPE KEY CAUSES REVERSION TO THE PRIOR ACTIVITY LEVEL

After you have helped the student set up the program, he or she can continue to play *BANK IT* with a minimum of supervision by pressing:

Esc

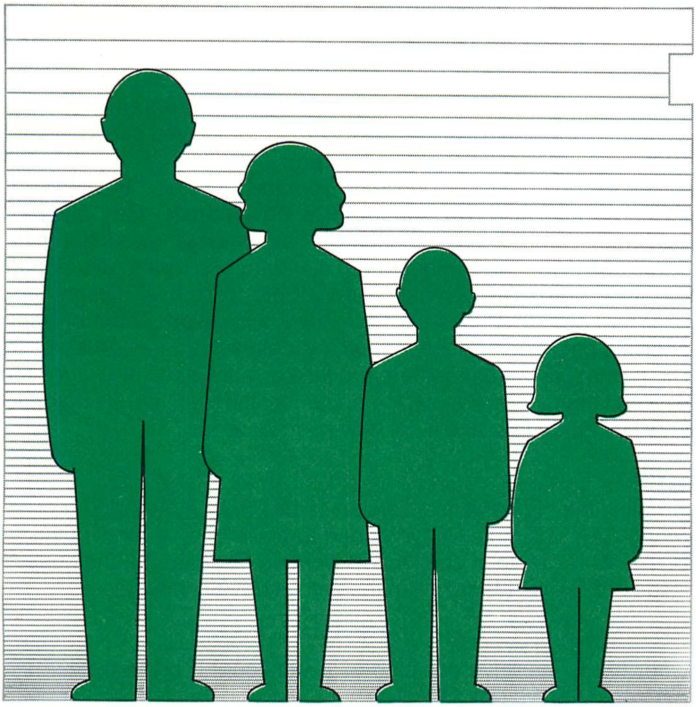
after each game to return to the main menu and then pressing:

↵

when the main menu reappears to start a new game. When you want to end the program, keep pressing the Escape key until you are back at the first question, the color question. Then press it one more time to go back one more step, and you will exit the program and be back in DOS.

ABOUT THE AUTHORS

BANK IT is one of 80 educational computer games distributed by Psychotechnics Inc., geared to the learning needs of children from kindergarten through eighth grade. As the programs were designed, they were carefully evaluated through pre-testing and post-testing of children in hundreds of school systems, mostly in California and on the West Coast. Further information, as well as packages of the programs themselves, are available directly from Psychotechnics Inc. We hope to offer additional selections from this superior library of educational software in future issues of *PC Disk Magazine*.



PERSONAL CASH FLOW MANAGER-PART III

By Gary Oppenheimer

Special Requirements: Personal Cash Flow Manager-Parts I & II
Files Used: CASH.BAS
CASH-RPT.BAS
CASH-BLD.BAS
TRANS.DAT
ACCOUNT.CDE
CHARGE.CDE
ARCHIVE.DAT

Parts I and II of the PERSONAL CASH FLOW MANAGER *provided a comprehensive cash tracking system for your personal finances. About the only thing missing was a facility to analyze your historical activity—reports such as variance from budget and year-to-year comparisons. This was no oversight. There are numerous software packages already on the market designed specifically to perform* *these kinds of analyses. So rather than reinvent the wheel, we decided to let you use your favorite such package by providing a facility to transfer your* PERSONAL CASH FLOW MANAGER *data to the package of your choice. Part III makes this transfer possible.*

BACKGROUND

Before we are called to task let us properly qualify our claim regarding "the analysis package of your choice." First and rather obviously, the analysis package must be able to read in data from an external file (that is, a file not created by the package itself). To transfer data, the *PERSONAL CASH FLOW MANAGER* program creates a file containing your charge category data, which you subsequently instruct your analysis package to read.

The second qualification stems from the fact that there is as yet no universally accepted format for such a transfer file—different packages demand that the incoming data be organized differently. The closest thing to a widespread standard is a format developed by Software Arts Inc., makers of VisiCalc, called DIF (Data Interchange Format). This is the format we have used. Therefore any analysis package capable of loading data from an external file in DIF format can be used (there are far too many such packages for us to list them here). Check the documentation for your package to see if it can do this, and if so, how to do it.

THE TRANSFER FILE IS IN DIF FORMAT

USE

Creating the third and final version of the *PERSONAL CASH FLOW MANAGER* is simple. All you need to do is copy the new CASH-RPT.BAS file to your Version 2.0 working diskette. Put the current issue of PC Disk Magazine in the default drive and type:

```
COPY CASH-RPT.BAS B: ↵
```

where B: is the drive containg your Version 2.0 copy of the *PERSONAL CASH FLOW MANAGER.*

Everything works exactly as before, with the addition of one new facility. From the system's main menu select choice 5 to go to the Report/Inquiry Submenu. On this submenu select choice 8 to begin the creation of your transfer file.

You can accept the default filename of CASHTST.DIF for your transfer file by pressing the Enter key, or you can type in a filename of your choice. Similarly you can accept the default range of charge categories to transfer, which is all of them, by pressing the Enter key in response to the computer's prompts for starting and ending category numbers. Or you can enter your own value for either to select a subset of all your data.

When these three parameters are set, the program will proceed to create a file with 24 months of actual and budgeted charge category data, for the charge categories selected. It will tell you as it does each month. Upon completion you are returned to the Report/Inquiry Submenu. You should then exit the *PERSONAL CASH FLOW MANAGER,* invoke your analysis package, read in the external DIF file created, and then begin analysis of your historical cash activity.

UTILITIES & DIAGNOSTICS

SORT/MERGE I

By Louis J. Cutrona

Special Requirements: None
Files Used: SORT.EXE
MERGE.EXE

SORT/MERGE I *enables you to sort a standard (i.e., ASCII) text file, as well as merge two sorted text files into a third sorted text file. The SORT and MERGE functions are separate processes (in fact they are separate programs). Both, however, are invoked with a single command, and both use the same command line syntax to specify the desired collating sequence for output. The sort function lets you specify up to three ''fields'', that is, three groups of characters within each line, by which to reorder the file. Each field can be sorted in ascending or descending sequence. The MERGE function will take any two text files sorted in the same sequence, and combine them into a third file sorted in that sequence. Together the SORT and MERGE programs can put any text file in any sorted order you want.*

TEXT FILE BACKGROUND

SORT/MERGE I is designed to work on what we call text, or ASCII, files. Text files consist of records (i.e., lines) of varying lengths (up to a maximum of 256 characters each). In most cases, each line is composed of ordinary, visible characters—letters, numbers, and various symbols—although special control characters, such as a tab, can also be present. Regardless of its contents, every line in a text file ends with a "carriage return," itself a special control code sequence that separates one line from another. This delimiter is what *SORT/MERGE I* uses to determine where one line ends and the next begins.

SORT/ MERGE 1 WORKS ON TEXT FILES

The other distinguishing feature of a text file is that each byte in the file represents a single character, and conversely every character in the file is represented by a single byte. The character each byte represents is determined by an arbitrary computer industry convention called ASCII, which maps every arithmetic value of a byte, computed in binary, to a character (see Appendix G of your IBM BASIC manual). Thus the terms *text file, ASCII file,* and *character file* refer synonymously to a file wherein each byte is a unit of meaning. To determine the sequence in such a file, *SORT/MERGE I* compares fields one byte at a time. The determination of sequence is based on the equivalent binary values of the characters compared (commonly known as the ASCII collating sequence). When a shorter record is compared to a longer record, the shorter record is temporarily extended with binary zeros to the length necessary for a byte-by-byte comparison.

Examples of text files are files created with a system editor like EDLIN (not created with a word processor). The MEASURE.DAT file in this issue is an example.

We emphasize that *SORT/MERGE I* is for text files only, so that you will avoid using it on the other major type of file—fixed-record files. All records are the same length in such files, but there are no delimiters between them. Furthermore, numbers are usually stored in special forms requiring several bytes, so byte-by-byte comparisons are meaningless. We will publish *SORT/MERGE II* in our next issue to handle fixed-record files.

SORT BACKGROUND

The SORT program uses a partition-exchange sorting algorithm (Quicksort) that is highly efficient in sorting large numbers of records. Quicksort makes use of the programming technique called *recursion,* discussed in a separate article in this issue.

Sorting is done byte by byte in the normal ASCII collating sequence. Tab characters are treated literally, not as a string of blank characters, and capital letters are sorted separately from lowercase letters. The order of records with equal keys is not necessarily preserved. Up to three sort fields may be specified. If no sort field is specified, the records will be sorted over their entire length. When sorting is to be done over one or more fields, the position of each field within the record must be specified. This is accomplished by indicating the location of the first character in the sort field (counting from the first character of the record, which is always in location 1), as well as the length of the sort field (the number of characters to be compared). In addition, the SORT command allows you to specify whether the sort fields are to be arranged in ascending or descending order.

Since workspace is needed in memory for the sorting process, the file to be sorted cannot be larger than 60K bytes. We will discuss a technique to create larger sorted files when we discuss the MERGE function.

SORT COMMAND SYNTAX

The SORT program is activated by a single command. The general format of the SORT command is:

```
SORT infile outfile / sort-field1/ sort-field2/ sort-field3
```

where sort-field is further subdivided into three parameters, separated by spaces:

```
/starting-location length sequence/
```

The following items are mandatory:

- The keyword SORT
- The infile specification
- The outfile specification

If no sort fields are specified, each line will be used in its entirety to sequence the file. When specifying a sort field, you must enter a starting location. If you do not also specify length, the entire line from starting location on will be used for sorting. If a sequence is not specified, the field is sorted in ascending order. Enter r or R to specify a sequence in reverse (descending) order.

The infile is the unsorted input file; the outfile is the sorted output file. The three sort fields, separated from the rest of the program invocation by slashes, specify the one, two, or three fields over which the sorting is to take place.

SORTING IS DONE WITH A SINGLE COMMAND

A typical example of program invocation is given below:

```
SORT FOO.DOC B:FOO2.DOC
/1 10 R/ 25 2 / 20 5 ↵
```

Explanation: The name of the unsorted input file is FOO.DOC on the default drive. Its records will be sorted over three sort fields. The first sort field begins in position 1 and has a length of 10 characters. The second sort field begins in position 25 and has a length of 2 characters. The third sort field, beginning in position 20, has a length of 5 characters. Records will be sorted into reverse (descending) sequence over the first sort string, and ascending sequence over the second and third sort strings. The sorted records will be written to a file named FOO2.DOC on drive B.

MERGE BACKGROUND

The MERGE program will merge two ASCII files in the same sorted order together to form a third, sorted ASCII file. During the merge, comparison of merge fields will be done byte by byte, and the merge will reflect the normal ASCII collating sequence.

There is no limit on the size of the input files to be merged, other than the availability of sufficient disk space to hold the resultant merged file. There are two requirements, however, if merging is to work properly. First, the two input files to the merge must be sorted by the same sort fields; that is, they must be in the same sort order. Second, the merge fields specified in the merge must be the same as the sort fields by which the input files are ordered. The merge produces an output file that preserves the sort order of the input files. To produce a sorted output file, the merge function must know which fields to compare when deciding whether the next output record should come from input file # 1 or input file # 2. The merge fields specify the appropriate comparisons to make. To work, the comparison must be between fields that are already in sorted order.

The MERGE function provides a ready way to build large sorted files, since files of any size can be merged. To build a large sorted file, begin by breaking it into pieces that the SORT function can handle. Arrange each piece in the same order. Then use the MERGE function to combine each piece into a cumulative output file. Since the output of a merge is in the same sorted order as the input, this output file can be repeatedly used as input in each step of the aggregation.

MERGE COMMAND SYNTAX

The MERGE command operates the same way as the SORT command. The general format for the MERGE command is:

```
MERGE infile1 + infile2 outfile /merge-field1/merge-field2/
  merge-field3
```

where infile1 and infile2 are the two sorted files to be merged; outfile is the merged output file; and the optional merge fields are separated from the rest of the parameters by a slash.

ONLY AVAILABLE SPACE LIMITS THE SIZE OF FILES TO BE MERGED

Up to three merge fields may be specified to control the merge. If no merge fields are specified, records will be merged based on their entire length. When merge fields are specified, their specifications follow the same format as the sort field specifications in the SORT command. First, the location of the first character in the merge field is given as an integer, with the count based on the first character in a record occupying location 1. Second, the length of the merge field (the number of characters in the field) is given as an integer. And finally, an indication is given of whether the merged records are to be arranged into ascending (default) order or descending (use "R" or "r") order with respect to the merge field.

A typical example of program invocation is given below:

```
MERGE MRG1.SRT + MRG2.SRT
FOO.SRT /1 10 R/25 2/20 5 ↵
```

Explanation: The two sorted input files to be merged are MRG1.SRT and MRG2.SRT on the default drive. Records from these two files will be merged to maintain the order of three merge fields. The first merge field begins in position 1 and has a length of 10 characters. The second merge field, with a length of 2 characters, begins in position 25. The third merge field begins in position 20 and has a length of 5 characters. In the merged file, records will be merged into reverse (descending) sequence over the first merge field, and ascending sequence over the values in the second and third merge fields. The file created by merging these two files will be named FOO.SRT on the default drive.

RECURSION

By Louis J. Cutrona, Jr.

The ASCII file sort program included in this issue employs an algorithm (a fixed procedure for computing a desired result) known as "Quicksort". This algorithm in turn is based on the use of a programming technique called *recursion*. *Recursion* occurs when a program or subroutine *calls itself* in order to perform its task.

When a program calls itself recursively, it must first save a record of where it is in its computations before actually carrying out the call. This is necessary so that when it returns from the call, the program "remembers its place" and can continue its calculations from where it left off.

The Quicksort algorithm sorts a list as follows:

1. If the list has just one member, there is no sorting to be done, so control returns to the calling program.
2. If the list has more than one member, a member of the list is picked at random (call it *X*), and the list is then rearranged (sorted) so that all of the list elements that are less than *X* are moved to the left of *X*, and all of the list elements that are greater than *X* are moved to its right. *X* is now in the position it will finally occupy when the list is completely sorted.
3. If there is at least one element to the left of *X*, Quicksort is called again to sort the list composed of all of the elements to the left of *X*. This calling process keeps recurring until all the elements to the left of *X* are in sorted order.
4. If there is at least one element to the right of *X*, Quicksort is called recursively as before to sort the list composed of all of the elements to the right of *X*.
5. The list is now completely sorted. Control is returned to the calling program.

IN RECURSION, A ROUTINE CALLS ITSELF TO PERFORM ITS TASK

Notice that each time a call is made to Quicksort, Quicksort may initiate up to two additional recursive calls to itself. Note however that each time Quicksort calls itself recursively, it calls itself to sort a list shorter than the list it is currently sorting. This guarantees that eventually Quicksort will call itself with a list containing only one item. When that happens, the call will return without making any further calls. Thus, Quicksort will not keep calling itself forever—it will eventually return with the list sorted in order.

Any useful recursive algorithm must have this characteristic: it must eventually stop making additional calls to itself. Otherwise, it will continue endlessly and, since each additional call requires additional storage to keep track of conditions when the call was made, it will eventually use up all available memory.

The programming device which saves program variables and status information (the program "state") prior to a recursive call and restores that information after the call returns is called the *pushdown stack*. A *pushdown stack,* also known as a Last-In/First-Out (LIFO) stack, is a memory-management method rather like a basket for papers on your desk constructed so that you can only add to or remove papers from the top of the stack.

Each time a recursive call is to be made, the current program state information is placed ("pushed") onto the top of the stack. When the call returns, the information on top of the stack is removed ("popped") and used to restore the program to its former state so that processing can resume from where it left off.

THE STACK OF PROGRAM STATES MAKES RECURSION POSSIBLE

The purpose of a stack is that each time a recursively called program calls itself, it can push its state information on top of the stack and pop it off later without disturbing any previous information on the stack. Since calls and returns are always paired, it doesn't matter (unless you run out of memory) how many additional recursive calls take place after a particular call pushes its data onto the stack. That data will always be on top of the stack when its call returns.

Another example of a recursive procedure is the factorial function. Factorial of (X) is defined as follows (X must be a positive integer):

1. If X is 1, then return the value 1.
2. Otherwise, return the value of X multiplied by the value of factorial (X-1).

Again, note that each time factorial calls itself, it calls itself using a smaller argument, so that eventually it will call itself with the value 1 and stop generating further calls.

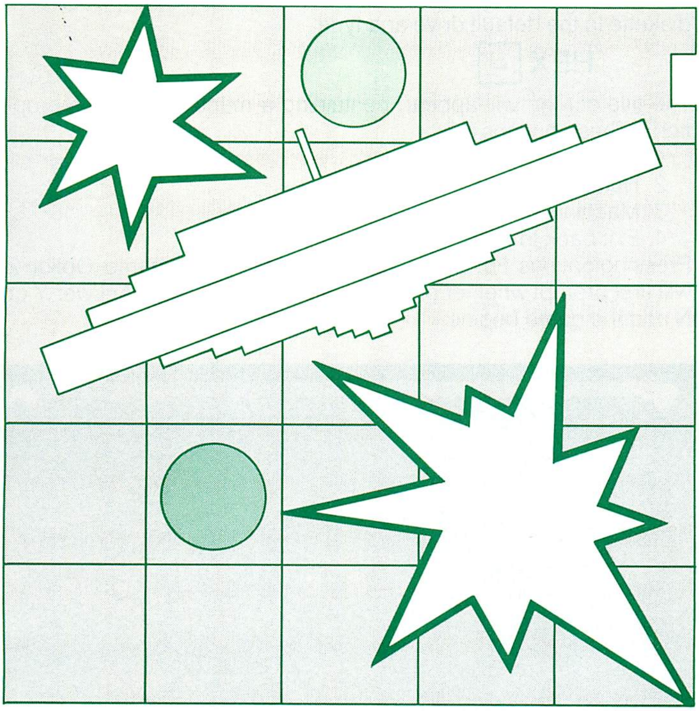
It is easy to see that you could write a program to return the same factorial values without using recursion at all. For example (again, X must be a positive integer):

```
1. N = 1
2. For I = 1 to X
3. N = N * I
4. Next I
5. Return the value of N
```

Here, we have substituted an iterative (For/Next) loop for the recursive calls in the previous algorithm. As it turns out, any recursive procedure can be transformed into a non-recursive procedure by the strategic introduction of iterative looping. The appropriate transformation is by no means always as simple as in our factorial example. Moreover, it is often more difficult to follow the logic of the non-recursive form of the procedure.

The decision to implement a recursive algorithm as a recursive procedure or as an iterative procedure generally depends on a number of factors, the most obvious being whether or not the programming language in which the algorithm is to be implemented supports recursion easily.

Recursion can be a powerful technique. More of its potential will bear fruit in the field of artificial intelligence, where much of the advanced work currently being done depends on the recursive abilities of programming languages like Pascal, C, PL/I, and especially LISP.



HEX

By Neil Sarnak & Isaac Dimitrovsky

Special Requirements: None
Files Used: HEX.EXE

Similar in form to the ancient games of Chinese Checkers and Go, HEX *is a procedurally simple game offering considerable strategic challenge. The playing area is an 11-by-11 grid of adjoining hexagons (called hexes) in the shape of a diamond. Your goal is to make a continuous path from one side of the diamond to the opposite side. Your opponent is seeking to do the same with the other two diamond sides. There's no race involved since only one player can win—it's pure strategy.*

Each player alternates placing their marker onto open hexes in the playing field. Block your opponent's line, and he can't win. Or can he? Keep in mind that all hexes (except the borders) allow connections in six possible directions.

Moves can be offensive or defensive, though the best moves *combine both qualities. Just make sure that you don't stretch your line too thin. Some of those position markers can get pretty isolated out there.*

THE "BOARD" IS A GRID OF HEXAGONS

GETTING STARTED

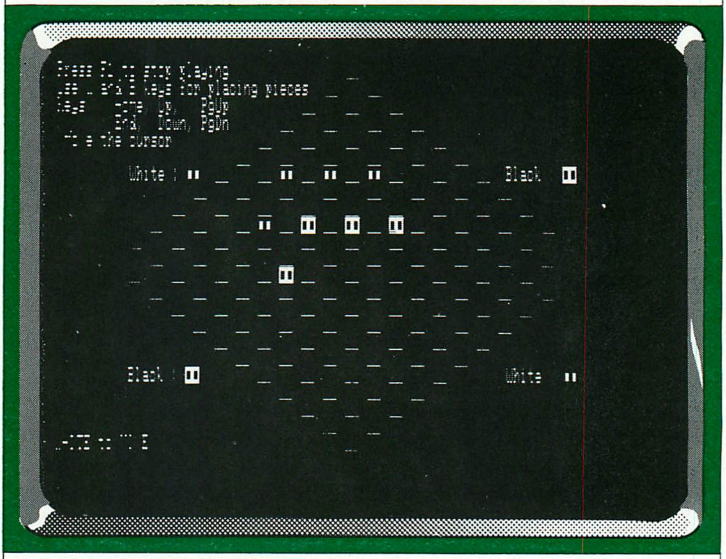
To start *HEX*, you must be in DOS. Place the *PC Disk Magazine* diskette in the default drive and type:

HEX ↵

The title screen will appear, containing a menu of four game options. These options are:

1. Human versus Human.
2. Human versus Machine.
3. Machine versus Machine (demo).
4. Exit back to DOS.

Pressing options 1 or 3 will take you directly into the game. Option 2 will first prompt whether or not you want to play white. Answer Y or N and the game begins.



```
Press F1 to stop playing
Use 1 and 2 keys for placing pieces
Keys   Home, Up,   PgUp
       End,  Down, PgDn
move the cursor
```

The game screen—flanking maneuvers in progress.

PLAYING THE GAME

After you've picked which of the three game options you prefer, a diamond-shaped grid of hexagons will form on your screen. This is the playing field, made up of 11 hexagons on each side of the diamond. The screen will also show the game's commands, which are discussed below.

Players alternate moves. White always moves first, starting game play. A prompt line below the playing grid indicates whose turn it is to play a piece.

The White player must create an unbroken string running from the top left of the grid to the lower right. Black must join the top right of the field to the lower left. The direction you go in, the number of convolutions your line makes, and what order you place your markers in do not matter. To win, you must be able to trace out a fully joined string of marked hexes running from one side of the diamond to its opposite side. Two identically marked hexes are considered joined only if they share a side in common. To mark a hex with your color, first position the cursor within it.

You position the cursor using only the top and bottom rows of your keyboard's numeric keypad. That is, keys

7 Home, 8 ↑, and 9 PgUp

are used to move the cursor up and to the left, straight up, and up and to the right, respectively. Similarly, keys

1 End, 2 ↓, and 3 PgnDn

are used to move the cursor down and to the left, straight down, or down and to the right, respectively. If you visualize the current cursor position as represented by the 5 key on the numeric keypad, the other number key positions reflect the corresponding paths through the sides of the vertically-oriented hexagons in the grid.

Once you've reached the hex in which you wish to place your marker, press W (for the WHITE player) or B (for the BLACK player) and the appropriate white or black marker piece will appear in that space.

Hitting any keys other than those noted above (other than the F1 (Exit) key) will not affect game play, but will sound a beep.

For convenience, the cursor always remains at the last move made by a player (the last hex occupied).

LOCATE THE CURSOR THEN MARK YOUR HEX

ENDING THE GAME

The program can detect when a player has won, and will declare so at the lower left of the game screen. The program will then pause, so that you can review the completed game board. You may then press any key to return to the title screen menu. It is from the title screen that you exit the game and return to DOS by entering:

4

as your choice from the main menu.

You can interrupt the game at any time by pressing:

F1

When you do so the computer will ask if you wish to end the game in progress. If you type Y for yes you will be returned to the title screen menu. Typing any other key will cause play to resume.

STRATEGY

Regardless of whether you're playing White or Black, playing against a human opponent or against the computer, your best basic tactic is to always place your markers in hexes with at least two adjoining hexes open in between, thus allowing the marked hexes to be joined in either of two directions. If your opponent attempts to hinder your line by occupying one adjoining hex, a connection to the rest of your line can still be made via the remaining free hex.

Remember, at some point or other on the playing field the two lines must cross each other. Once your line has been breached, you have little choice but to begin a new line where it can intersect (and block off) your opponent's string of markers. The first player to block off an opponent's access to his target side usually wins.

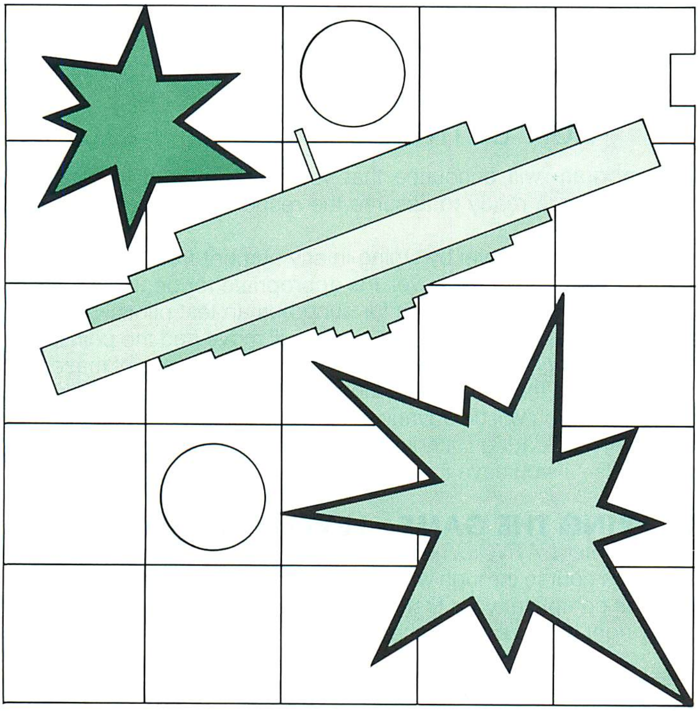
But don't take up time running your line along one of your opponent's sides. All your opponent needs to do is occupy one hex on that side and your efforts are wasted!

BLOCK YOUR OPPONENT WHILE EXTENDING YOUR LINE

One good starting strategy is to place your first marker at a point one hex away from the diamond apex (the four points at which the White and Black sides meet). Be sure to give yourself at least two paths to eventually connect to that side. Continue positioning markers at points where there are at least two hexes to connect to your previous pieces. Then pick a point near the center of the playing field where you wish to intercept your opponent's route, and position pieces to either side of this point. When you've reached a point where you have two hexes by which you can reach your target side, go back and fill in the gaps in your line.

A good variation on this strategy is to create a series of "W's" with your pieces spaced one hex away from each other, progressing toward your opposite side. Once completed, all you will need to do to win is to go back and fill in the gaps. But remember, if you're blocked at any point along the string where there is only one possible connection, you are in trouble.

So, regardless of your playing strategy, watch your opponent!



DATNOIDS

By Casey Roche

Special Requirements: None
Files Used: DATNOIDS.BAS

Fasten your seat belts, extinguish all smoking materials, and prepare to accelerate through hyperspace. You're about to become the pilot of a DATNOIDS *spaceship, and you will have to steer your craft skillfully and fire on obstacles with a steady hand as you speed through space towards THE FINAL TEST. But beware! Exceed the limitations of the hyperdrive or collide with an antimatter wall, and both you and your ship will be consumed in an abrupt, cataclysmic explosion.*

BACKGROUND

DATNOIDS is an arcade-style game built for single-player operation. You must manuever a spaceship through three mazes and fire missiles at barriers blocking the interstellar pathway. Your ship is destroyed when you hit a wall or a barrier or when a missile you fire ricochets back at you. You get five ships at each level of difficulty and you accumulate points as you make your way through space.

GETTING STARTED

To start *DATNOIDS*, load Advanced BASIC into your PC by typing:

```
BASICA ↵
```

The program needs only a disk drive and a monochrome monitor. So, if you think you're ready, place the *PC Disk Magazine* diskette into your default drive and type:

```
RUN"DATNOIDS ↵
```

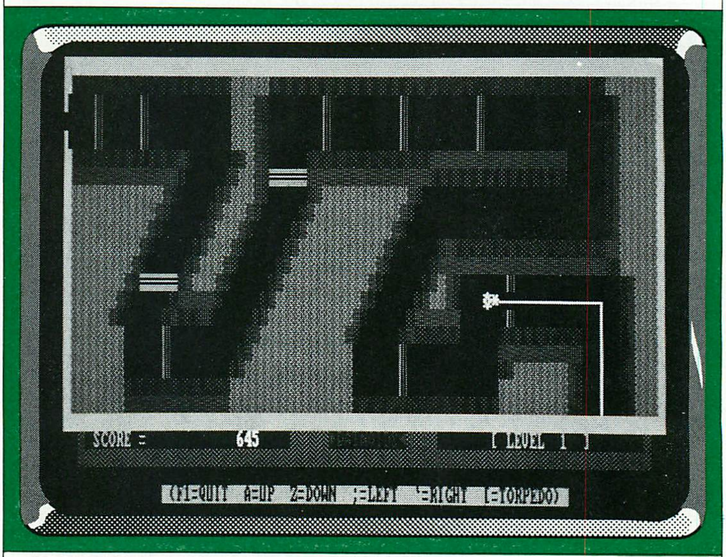
ENTER YOUR SKILL LEVEL BY TYPING IN ANY NUMBER FROM 1 TO 10, THEN HITTING THE ENTER KEY

The program will announce that you have entered the Datnoid Realm and are ready to assume the responsibilities of a Datnoid Pilot.

Enter your skill level by typing in any number from 1 to 10, then pressing the Enter key. Level 1 is appropriate for beginners, while level 10, the highest level, is for superhuman test pilots. The level controls the speed at which your ship will move and the points you will receive for each manuever. Then enter your choice of maze levels. Level 1 is the easiest and the level 3 maze is the most difficult. As you win, you will be advanced to the upper levels automatically. However, the daring Datnoid Pilot may proceed directly to the upper levels—if you have the nerve.

PLAYING THE GAME—LEVELS 1 AND 2

As you enter *DATNOIDS* on level 1 or level 2 the program will construct the course through which you will have to navigate. When the maze is complete, your first ship will appear on its launcher at the lower right hand corner of the screen. Your path ends in the upper



```
SCORE =          645                    [ LEVEL 1 ]

(F1=QUIT  A=UP  Z=DOWN  ;=LEFT  '=RIGHT  [=TORPEDO)
```

Keep a light touch on the controls

left hand corner where there is an opening in the wall through which you may fly out of the maze. Along the way are antimatter barriers which you must pierce with missiles in order to pass through.

You first will hear the sound of the engines of your ship charging up to launch power. Three beeps will signal the countdown, then you'll blast off. Your *DATNOID* ship can move up, down, left, or right using the following keys:

A – UP

Z – DOWN

; – LEFT

‘ – RIGHT

To fire missiles (of which you have an inexhaustible supply) press:

[

Note that this allows you to turn the ship left and right with the index and ring fingers of your right hand and fire missiles with your middle finger. This list of keystroke commands appears at the bottom of each game screen. Your current score, the level of play, and the skill level (speed) are also constantly displayed.

Fly carefully and fire at will. Colliding with a wall or a barrier will be hazardous to your extra-terrestrial health. And don't fire wantonly. Errant missiles have a nasty habit of bouncing off the antimatter walls and coming back to destroy you. And you can't outmanuever them, since it takes all of your ship's power to fire the missiles.

THE LIST OF KEYSTROKE COMMANDS APPEARS AT THE BOTTOM OF EACH GAME SCREEN

Treat your hyperdrive power unit gently. If you command it to go in a direction in which it's already traveling or try to abruptly reverse the direction of the ship you will push the drive beyond its capacities and destroy your ship. After the explosion, the charred walls of the maze are repaired, the barriers reformed, and your next ship begins the countdown procedure back at the launcher. Losing all five ships ends your mission.

If you successfully navigate the course at one level, you will be accelerated via a hyperspace jump to the next level of play, where you will receive five new ships.

PLAYING THE GAME—LEVEL 3

If you succeed in reaching level 3 (or are foolhardy enough to start there) you will see that enemy ships have deposited mines across the screen in a dense random pattern. You must navigate around these because a collision with one will destroy your ship. However, the Datnoids Fleet maintains three secure areas in this sector, and you will see these form on the screen and clear the mines within them. To get in or out of these areas you must blast a hole in the force field wall with a missile. Once these areas have been formed, your launching sequence begins. This level is the final and most difficult test for a Datnoid Pilot. Few emerge victorious from its trials. If your ship is destroyed, the enemy will rearrange the mines in a new pattern for your next attempt.

If you succeed in mastering The Final Test, congratulations. But don't rest on your laurels too long. You are soon accelerated through hyperspace back to level 1 of *DATNOIDS* and given a faster ship. You must begin the series of challenges again.

Like any champion, the victorious *DATNOIDS* pilot must always defend his standing against new challenges. The only end is in final, inevitable defeat at an ever-higher level of play. Good luck!

SCORING AND STRATEGY

Points are awarded in *DATNOIDS* for collision-free navigation, for successfully firing the missiles, and for completing a level. The faster the speed you choose as your skill level, the more points you will receive for each maneuver. Once you have lost all five ships at any level, the game is considered over and your final score is displayed. The score is then reset to zero for the next player.

A *DATNOID* pilot will soon discover a few secrets: the craft can fire many missiles in a row, and they will all travel along exactly the same path. This salvo approach is useful if you are headed towards several consecutive barriers. The second torpedo will travel through the hole made in the barrier by the first torpedo and continue on to explode on the second barrier. If you do this, be careful. Fire too many shots and the last will richochet off the far wall, travel back through the punctured barriers, and destroy you.

THIS LIST OF KEYSTROKE COMMANDS APPEARS AT THE BOTTOM OF EACH GAME SCREEN

Since the *DATNOIDS* ship can steer only vertically and horizontally, the diagonal passageways are particularly treacherous. They must be navigated with a series of quick turns. Only experience can teach you how close to come to a wall before turning away on the next tack.

In level 3 *DATNOIDS*, the path from the launch site to the upper right safety force field is always clear of obstacles, no matter how the rest of the board has been arranged. The pilot can use the time spent traveling along this safe starting path to scan the rest of the screen for the best route to the goal portal.

ENDING THE GAME

DATNOIDS will keep challenging you at successively more difficult speeds until it finally defeats you. If, however, you do not wish to go down in a blaze of glory, pressing the F1 key at any time will end the program and return you to Advanced BASIC.

DATA FILES

MEASUREMENT DATA FILE

By Morris Effron

Special Requirements: None
Files Used: MEASURE.DAT
MEASURE.BAS

Remember when America was going to convert to the metric system? At last we were going to join the rest of the civilized world in the use of a rational and scientific system of measurement. Despite all the fanfare and promotion, the movement never really took hold. It's surprising how attached we are to an admittedly motley collection of standards for sizing things up.

We at PC Disk Magazine *would never presume to resurrect such a losing crusade, and this issue's data file contains no such ulterior motive. The fact is though that we Americans often want to convert measurements between the American and metric systems, espe-*

cially when travelling abroad (how much is 1.34 francs per liter of petrol?). Furthermore it is sometimes useful to convert measurements between units of the same system (how many acres in a square mile?).

The MEASUREMENT DATA FILE *contains a table of seven measurement categories. Within each category the relevant units of measure for both the metric and American systems are recorded, along with intra-system and inter-system scaling factors. The accompanying program reads this table, and enables you to find the equivalent amounts of any two units of measure in the same category. While this may not inspire you to take your PC along on your next vacation, you may still find the program useful and instructive.*

BACKGROUND

The name of the file containing the measurement conversion data is MEASURE.DAT. This is an all ASCII character file, so it can be printed or called into a text editor. The file is organized into seven contiguous category groups of three lines each. The first line of each group contains the name of the measurement category to which these three lines apply. Immediately after the category name are two numbers. The first is the number of different measurement units in this category under the metric system. The second is the number of measurement units under the American system. So for example the category headed "Weight, 3,3" says that within the weight measurement category there are three different units of metric weight (metric ton, kilogram and gram), and three different units of American weight (ton, pound and ounce).

FOR EACH SYSTEM ONE UNIT OF MEASURE IS GIVEN A REFERENCE VALUE OF 1

These counting numbers are not essential, but facilitate the program's reading of the following two lines for each category. The next line lists the metric units of measurement in the category, and the subsequent (and last) line lists the American units. These two lines contain pairs comprised of a measurement unit name followed by a number. The purpose of the unit name is obvious enough, but what about the number?

In order to compute the equivalence between any two units in a category, two pieces of information are necessary. First, all the units within a given system of measurement (metric or American) must be correlated with a common reference value, so that conversions between them can be performed. Thus for each system in each category, one unit of measurement is given a reference value of 1, and all the other units are listed with their associated multiple of this reference value. For example in the length category, the metric reference unit is the centimeter, which is followed by a 1. A kilometer is 100,000 centimeters, so it is followed by 100,000. At the other end a millimeter is equal to .1 centimeters, so the value .1 appears following it. This approach makes conversion of related units easy.

This approach also has another critical benefit, which pertains to the second requirement for conversion calculations. To calculate equivalences between the units of the two different measurement systems, we need a converion factor. A highly inefficient solution would be to include a conversion factor for every combination of inter-system units—not a very clever approach. For a given category all we really need is one conversion factor: the equivalence between the reference units of the two systems. Since all other units can be converted to and from their reference unit, the conversion factor for the reference units is the bridge for all other units in the category. Hence the last number in the list of American units within each category is the number of metric reference units equivalent to one American reference unit. With this arrangement of the data, the mathematics of measurement equivalence become quite simple.

You will notice a certain amount of data redundancy in the file MEASURE.DAT (e.g., Dry Capacity, Cubic Capacity and Liquid Capacity all use the same metric units of measurement). The reason for this is the benefit of treating all measurement categories in a uniform way. By so doing our programming effort is simplified, and the logic of our program is much clearer. Storage efficiency is not always a design goal above all others.

STORAGE ECONOMY IS NOT ALWAYS THE PREEMINENT FILE DESIGN GOAL

OPERATION

To use the measurement conversion program MEASURE.BAS, begin by loading Advanced BASIC into your PC. Put a system diskette with BASICA.COM on it in the default drive and type:

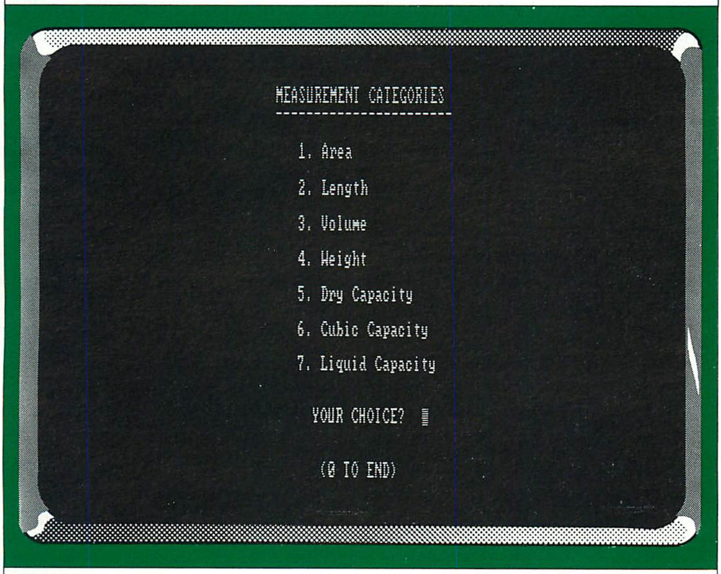
```
BASICA ↵
```

Then put your *PC Disk Magazine* in the default drive and type:

```
RUN"MEASURE ↵
```

The program will first tell you that it is reading the measurement data table, and then present you with the main menu of seven measurement categories to choose from.

Choose the category for conversion calculations by simply entering the associated category number followed by the Enter key. To exit



```
MEASUREMENT CATEGORIES
----------------------

   1. Area

   2. Length

   3. Volume

   4. Weight

   5. Dry Capacity

   6. Cubic Capacity

   7. Liquid Capacity

     YOUR CHOICE? ▌

      (0 TO END)
```

The main menu—enter your choice of measurement category.

the program simply type:

0 ↵

when you are at the main menu.

Upon selecting a proper measurement category from the main menu, you will be presented with a second screen displaying the measurement units of that category. The left-hand column, listed with capital letter references, are the metric system units. The right-hand column, with numeric references, are the American system units. At the bottom of the screen is the universal structure to be used to formulate your conversion request:

Amount ______ **Type** ______ = **???????** ______ **Type**

THERE IS A UNIVERSAL CONVERSION REQUEST FORMAT

To make your request, all you have to do is fill in an amount (i.e. some quantity), the reference letter or number specifying the units of this amount, and lastly the reference letter or number specifying the units into which you want this amount converted. The reference letter or number is what belongs in the Type field shown above. The amount to be converted can be any value from .000001 (10E-6) to 1,000,000 (10E+6). Enter it without commas. The one character type code can be any letter or number associated with a unit of measure in the list on display for the category. Conversion can be done between units of different measurement systems, or between units of the same measurement system.

Complete your entry for each of the three fields by pressing the Enter key. If you have specified an invalid value the field will be blanked out, without a message, and the cursor repositioned at the beginning of the field for another entry. When the third field is properly entered, and your request thereby completed, the program will automatically compute and display the expanded conversion equivalence. This display will remain on the screen until you press any key to begin entry of another conversion request.

To exit from a specific measurement category and return to the main menu, just type:

0 ↵

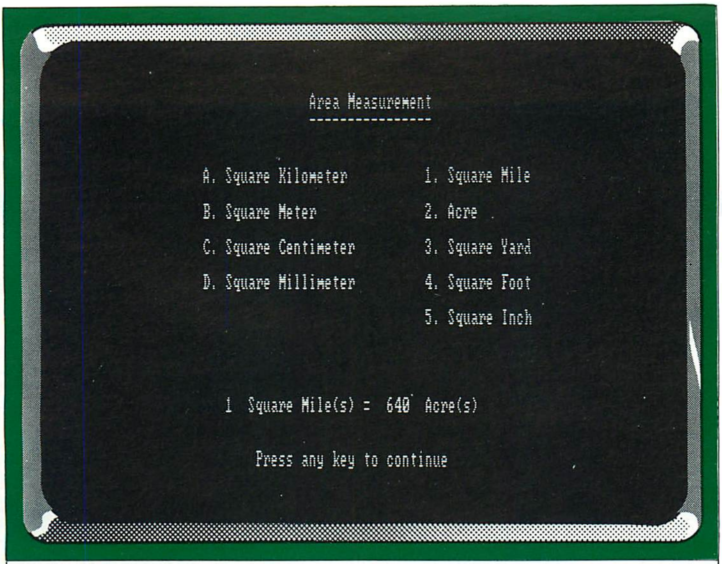
in the amount field.

Two quick examples should help illuminate this procedure.

Suppose you want to know the number of acres in a square mile. From the main menu enter:

1 ↵

to select the area measurement category, which contains the units acres and square miles, for conversion processing. Of course this presupposes that you already know which units of measurement are in which measurement categories. There's no reason that you should know this off the top of your head, but all you have to do is select each measurement category in turn to be shown which units are in which categories.



```
                 Area Measurement
                 ----------------

   A. Square Kilometer          1. Square Mile
   B. Square Meter              2. Acre
   C. Square Centimeter         3. Square Yard
   D. Square Millimeter         4. Square Foot
                                5. Square Inch


        1 Square Mile(s) = 640 Acre(s)

           Press any key to continue
```

A successful conversion of square miles to acres.

When the area measurement screen appears, note that square mile is type code 1 and acre is type code 2. Therefore you should formulate your request as follows:

1 ______ 1 ____ = ??????? 2 ____
Amount Type Type

This request can be read as "One square mile equals how many acres?" Press the Enter key after typing the value for each field. When you press Enter after the 2, the program will replace your request with the display:

1 Square Mile(s) = 640 Acre(s)
Press any key to continue

This is the expanded response to your encoded request. Now press any key and the request form will be redisplayed. Enter 0 in the amount field to return to the main menu.

WHEN YOUR REQUEST IS COMPLETE, CONVERSION IS AUTOMATIC

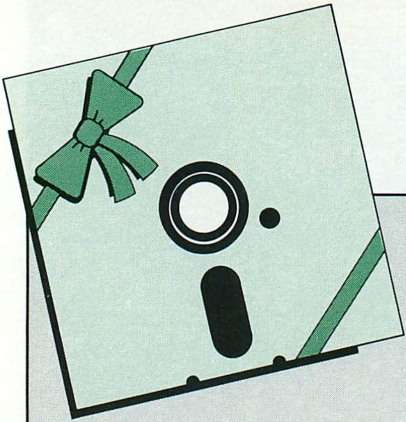
So much for conversion between units of the same measurement system. Performing an inter-system conversion works the same way. To find the number of quarts in 1.5 liters, select category 7 from the main menu. On the Liquid Capacity Measurement screen, formulate your request as follows:

1.5 ______ B ____ = ??????? 2 ____
Amount Type Type

The computer will reply:

1.5 Liter(s) = 1.5856 Quart(s)

That's all there is to it.



EXPAND YOUR SOFTWARE LIBRARY

This issue of *PC Disk Magazine* has already increased your software library for your IBM personal computer. If you haven't subscribed to *PC Disk Magazine*, just think of how easy it will be to expand your library with future issues delivered to your door at substantial savings. And if you already are a subscriber, wouldn't you like to share your knowledge with a friend and get them a gift subscription to *PC Disk Magazine*.

Either way, you can save $60 with this introductory offer and give yourself or a friend up to 72 programs for as little as $2.00 each! *PC Disk Magazine* is *the* solution to the problem of building and maintaining an up-to-date library of tested software programs.

You will get up to 12 varied and ready-to-use programs and files in every issue—complete with an illustrated user manual—for the ultimate in ease, efficiency and economy.

Every program in *PC Disk Magazine* is developed by experts, tested by the editors of *PC Disk Magazine* and published by the people who bring you PC, guaranteeing quality and satisfaction.

Use the enclosed card to subscribe today or to give a friend a gift. If the card is missing, please write to:
PC Disk Magazine, PO Box 2462, Boulder, Co. 80322
For even faster service, call our toll-free number below and use your American Express, MasterCard or Visa . . . today!

CALL
1-800-526-0666
for FAST service
and get
6 issues of
PC Disk Magazine
for only $119
a savings of 34%!

COMING UP

QUIKPRO+ (PART I OF II)

A program that writes BASIC programs. Describe your information to it, and programs are created to store and maintain that information (Part I), as well as report it (Part II).

PIE CHARTER

The third Euclidean element in our series of stand-alone, automatic graphing programs.

BIBLIOGRAPHER

Keep a bibliographic reference file of media items that interest you. Recall them by topic.

TRANSPARENT SORT

Learn a sorting technique (the Shell/Metzner sort) through a step-by-step presentation and explanation of an example in BASIC.

EARLY READER

A reading and spelling primer built on clear, vibrant displays.

SORT/MERGE II

The complement to this issue's text file utility; a pair of routines to operate on fixed-length record files.

ECONOMIC PIE

Put your economic convictions to the test in this game of pie shop management.

OENOPHILIAC DOSSIER—CHAPTER 1

The secret life of the grapes—a reference file of wine ratings.

CALLING ALL PROGRAMMERS!!

If you have written software for the IBM PC that you feel would be of use or interest to others, *PC Disk Magazine* would like to know about it. Publishing is no longer just for poets and prosaists, but now it's for programmers too. Ask for our Software Submission Plan by writing to:

PC Disk Magazine
Dept. 732
One Park Avenue
New York, N.Y. 10016

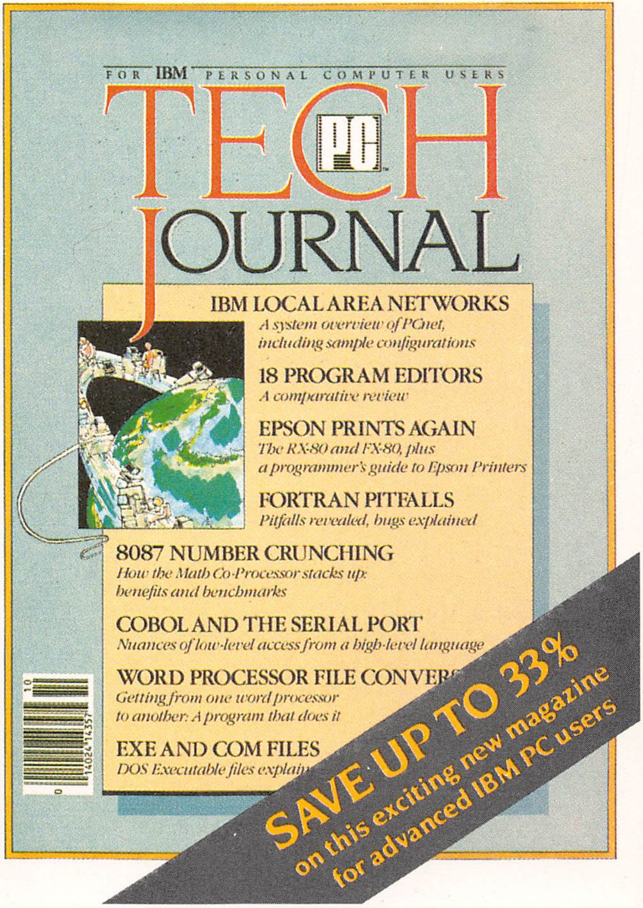
NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 66 BOULDER, CO

POSTAGE WILL BE PAID BY ADDRESSEE

TECH PC JOURNAL
P.O. Box 2966
Boulder, Colorado 80321



FOR IBM PERSONAL COMPUTER USERS

TECH PC JOURNAL

IBM LOCAL AREA NETWORKS
A system overview of PCnet, including sample configurations

18 PROGRAM EDITORS
A comparative review

EPSON PRINTS AGAIN
The RX-80 and FX-80, plus a programmer's guide to Epson Printers

FORTRAN PITFALLS
Pitfalls revealed, bugs explained

8087 NUMBER CRUNCHING
How the Math Co-Processor stacks up: benefits and benchmarks

COBOL AND THE SERIAL PORT
Nuances of low-level access from a high-level language

WORD PROCESSOR FILE CONVERS
Getting from one word processor to another: A program that does it

EXE AND COM FILES
DOS Executable files explain

SAVE UP TO 33%
on this exciting new magazine
for advanced IBM PC users

IF YOU'RE AN ADVANCED IBM PC USER, YOU'RE READY FOR PC TECH JOURNAL

PC TECH JOURNAL is the technologically sophisticated magazine on microcomputers, written for those who want to maximize the potential of their PC, as well as develop new applications and face new creative challenges.

PC TECH JOURNAL is the "Technical Gourmet" of computer magazines, offering you thought provoking and useful articles on how to fully exploit your system. You'll learn how to develop more elegant programming methodologies, get a better understanding of the inner workings of your PC and learn how to evaluate IBM protocols and standards.

If you are a serious programmer...if you work in MIS or DP...if you would rather write software than buy it...if you welcome technological adventure...YOU ARE READY FOR *PC TECH JOURNAL!*

Detach and mail the card today and take advantage of our discount program.

YES! SEND ME TECH PC JOURNAL FOR:

□ 1 year
(6 issues)
only $19.97—
SAVE 20%

□ 2 years
(12 issues)
only $36.97—
SAVE 26%

□ 3 years
(18 issues)
only $49.97—
SAVE 33%

Savings based on full one-year subscription price of $24.97.

CHECK ONE: □ Payment enclosed □ Bill me later

Charge To:

□ Visa □ MasterCard □ American Express

Card No.________________ Exp. Date________

Mr./Mrs./Ms.________________________

8H211

Company________________________

Address________________________

City____________ State__________ Zip________

Add $12 per year in Canada and all other foreign countries.
Please allow 30 to 60 days for delivery.

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 69 MORRISTOWN, NJ

POSTAGE WILL BE PAID BY ADDRESSEE

A LIBRARY OF SOFTWARE PROGRAMS FOR THE IBM PERSONAL COMPUTER
PC DISK
MAGAZINE
CN 1916
Morristown, New Jersey 07960

NO POSTAGE
NECESSARY
IF MAILED
IN THE
UNITED STATES

BUSINESS REPLY CARD

FIRST CLASS PERMIT NO. 69 MORRISTOWN, NJ

POSTAGE WILL BE PAID BY ADDRESSEE

A LIBRARY OF SOFTWARE PROGRAMS FOR THE IBM PERSONAL COMPUTER
PC DISK
MAGAZINE
CN 1916
Morristown, New Jersey 07960

Save $60.00
on

A LIBRARY OF SOFTWARE PROGRAMS FOR THE IBM PERSONAL COMPUTER

PC DISK

MAGAZINE

YES! Send me 6 issues of *PC Disk Magazine* for just $119.00—34% off the full subscription price of $179.00.

Check One:

□ I enclose $______________.
(Residents of CA, CO, CT, DC, FL, IL, MA, MI, MO, NJ, NY State, and VT add applicable sales tax.)

□ Bill me later.

□ Charge my: □ American Express
□ MasterCard □ Visa

Card No.______________________________
5000

Exp. Date______________________________

Mr./Mrs./Ms.______________________________
(please print full name)

Address______________________________

City______________________________

State________________Zip______________

Offer valid in U.S. and its possessions only. Please allow 30 to 60 days for delivery of first issue.

Save $60.00
on

A LIBRARY OF SOFTWARE PROGRAMS FOR THE IBM PERSONAL COMPUTER

PC DISK

MAGAZINE

YES! Send me 6 issues of *PC Disk Magazine* for just $119.00—34% off the full subscription price of $179.00.

Check One:

□ I enclose $______________.
(Residents of CA, CO, CT, DC, FL, IL, MA, MI, MO, NJ, NY State, and VT add applicable sales tax.)

□ Bill me later.

□ Charge my: □ American Express
□ MasterCard □ Visa

Card No.______________________________
5000

Exp. Date______________________________

Mr./Mrs./Ms.______________________________
(please print full name)

Address______________________________

City______________________________

State________________Zip______________

Offer valid in U.S. and its possessions only. Please allow 30 to 60 days for delivery of first issue.